# Sound compression – a rough path approach

Terry J. Lyons and Nadia Sidorova[1]
University of Oxford

**Abstract.** We present a new approach to sound compression, based on rough path theory, which turns out to be more effective than the traditional Fourier and Wavelet methods. We describe a procedure for encoding a signal using certain sequences of iterated integrals known as signatures and construct an algorithm for decoding.

**1. Introduction.** We take the view that perception of sound is an example of a controlled system: one has a complex dynamic system $Y_t$ (perception) subject to an external control $X_t$ (air pressure) where it is reasonable to hypothesize the existence of a model relating the internal state $Y$ of the system and the external control $X$ in the form of a differential system[2]

$$dY_t = f(Y_t)dX_t. \tag{1}$$

In order to accurately reproduce the sound, and still more to construct efficient compression algorithms retaining the essential information in the control $X$ one has to know the answer to the question: when do two sound streams $X$ and $\tilde{X}$ sound essentially the same (or two controls have the same effect)? This has a physiological component – but it also has a significant mathematical component that is, to an important extent, independent of the detailed structure $(Y, f)$ of the model.

This latter part is the focus of our paper. We believe that, if various mathematical challenges can be resolved adequately, a more systematic and less ad-hoc approach to the compression of continuous datastreams will emerge. Recent work [4] shows that a control $X$ is perfectly described through a series of algebraic co-efficients, based on computing iterated integrals and known as the signature of $X$. This transform can be thought of as a non-commutative analog of the Fourier series tuned to control problems. With even this result it would seem worth exploring the possibilities for encoding signals using their signature. There is a second complimentary reason.

On timescales corresponding to perception, the substantial flow of information in the control $X$ give it a complex, if continuous, structure (a spectral analysis of sounds suggests that it can be much rougher than Brownian motion). The theory of Rough Paths (see [2]), developed over the last years by one of the authors with several co-authors, enables one to give an exact mathematical meaning to the equation (1) even in cases where the path $X$ is much rougher than even a Brownian path. More importantly it provides ways to approximate to $X$ by more regular paths $\tilde{X}$ in ways that ensure, in a mathematically precise way "that the two signals $X$ and $\tilde{X}$ produce essentially the same response $Y$!".

Controlled systems exhibit an essential contrast in their behavior according to the dimension of the control variable.[3] In the case where $X$ is one-dimensional, the map $X \to Y$ is continuous in the topology of uniform convergence, i.e. it is robust to errors in $X$. On the other hand, if $X$ is multi-dimensional then this map is in general not continuous in the uniform topology. This instability underlines the essentially nonlinear nature of the control problem; summarising $X$ by its Fourier/Wavelet coefficients is limited in part because of the linearity of the Fourier approach. Small time shifts of the signal might be acceptable in one channel but in a multichannel datastream, independent compression that produced small time shifts in the reconstructed signals could prove a significant source of jitter.

---

[1]Mathematical Institute, 24-29 St Giles' Oxford OX1 3LB, UK Supported by EPSRC GR/R2962/8/01

[2]One might write: $\dot{Y}_t = f(Y_t)\dot{X}_t$ where $\dot{X}$ is the control and $Y$ is the response. However our interest is in cases where the controlling input $X$ is complex – and has no useful derivative.

[3]Similarly, differential systems have a lot more structure than single ODEs.

The two approaches compliment each other and are interrelated. They take full advantage of the multidimensional and non-linear aspects of the control problem.

In this paper we discuss the coding of a signal through its truncated signature and provide an algorithm for the decoding. It appears that, at the moment, availability of decoding algorithms is the main obstacle to the evaluation of the approach, which, on simple trial situations looks very promising.

**2. Compressing the signal.** Let $X_t$ be a path of bounded variation in $\mathbb{R}^d$, $d > 1$. For every time interval $[s,t]$ we define

$$\mathbf{X}^n_{s,t} = \int \cdots \int_{s<u_1<\cdots<u_n<t} dX_{u_1} \otimes \cdots \otimes dX_{u_n} \in (\mathbb{R}^d)^{\otimes n}$$

and we call

$$\operatorname{sign}_{s,t}(X) = 1 + \mathbf{X}^1_{s,t} + \cdots + \mathbf{X}^n_{s,t} + \cdots$$

the signature of $X$ over the time interval $[s,t]$. Further, we call

$$\operatorname{sign}^{[n]}_{s,t}(X) = 1 + \mathbf{X}^1_{s,t} + \cdots + \mathbf{X}^n_{s,t}$$

the truncated signature of $X$ at level $n$ over the time interval $[s,t]$.

Thus $\operatorname{sign}_{s,t}(X)$ is an element of the closure $T((\mathbb{R}^d))$ of the tensor algebra $T(\mathbb{R}^d)$ and $\operatorname{sign}^{[n]}_{s,t}(X)$ belongs to its truncation $T^{[n]}(\mathbb{R}^d)$. In the sequel we will skip the indices $s,t$ if the signature is taken over the whole time interval.

There are many dependencies between different iterated integrals $\mathbf{X}^n_{s,t}$ and so one has a lot of redundancy in the whole sequence $\operatorname{sign}_{s,t}(X)$. This superfluity of information can be avoided by taking the logarithmic signature and truncated logarithmic signature

$$\operatorname{logsign}_{s,t}(X) = \log(\operatorname{sign}_{s,t}(X)) \qquad \text{and} \qquad \operatorname{logsign}^{[n]}_{s,t}(X) = \log^{[n]}(\operatorname{sign}^{[n]}_{s,t}(X)),$$

where log and $\log^{[n]}$ is the usual and the truncated logarithm on $T(\mathbb{R}^n)$, respectively.

Now suppose one is trying to digitally approximate a continuous multi-dimensional signal $X$ where the interest is in using this approximation as a control in substitution for the original signal. It can be significantly more efficient to record the truncated logarithmic signatures $\operatorname{logsign}^{[n]}_{t,t+\delta t}(X)$ allowing satisfactory results on a relatively coarse time scale rather than record just the usual increments $X_{t+\delta t} - X_t$ where a much finer time scale would be required. More precisely, suppose that the signal $X$ has been measured on a very fine time scale $0 < t_1 < \cdots < t_{km} = 1$. Now it can be thought of as a continuous piecewise linear path. Let us combine the time intervals into the groups of $m$ intervals following each other and call them long time intervals. For the $i$-th long time interval, we compute the $n$-th truncated logarithmic signature $l^{[n]}_i$ of the piecewise linear path over this interval. Finally, we obtain the vector

$$l = l_{n,m,k}(X) = (l^{[n]}_1, \dots, l^{[n]}_k)$$

as a compression for the vector $(X_{t_1}, \dots, X_{t_{km}})$. The compression of information is achieved mainly because of the fact that it is sufficient to compute the elements of $l$ with much lower precision than the original data $X_{t_i}$.

**3. Reconstruction problem.** This approach provides a fast and simple algorithm for sound compression for a wide range of $n, m$. For the low dimensional choice where $n = 2$ and $m = 3$ the reconstruction problem of finding a stream $\tilde{X}_i$ such that $l_{2,3,k}(\tilde{X}) = l_{2,3,k}(X)$ is linear with only one quadratic constraint. So one can easily find $\tilde{X}$ satisfying the equation and even force the minimality of the length of $\tilde{X}$ over all solutions. Numerical experiments show how, even at this primitive level, $n = 2$ (using the groups of $m = 3$ intervals) the approach is effective and high quality[4]. The sound recordings made for the controls $X$ and $\tilde{X}$ show that although the streams $X$ and $\tilde{X}$ are not at all close on a sample by sample basis they produce the same acoustic effect when listened to with high quality headphones.

A fundamental difficulty in developing the approach seems to be that the problem of reconstructing a signal is much more difficult for $n > 2$. Let us formulate it precisely. Denote by $G^{[n]}$ (resp., $L^{[n]}$) the set of all elements which are $n$-truncated signatures (resp., logarithmic signatures) of paths of bounded variation.

**Problem:** Given $g \in G^{[n]}$, find a path such that $\mathrm{sign}^{[n]} X = g$ (or, equivalently, given $l \in L^{[n]}$, find a path such that $\mathrm{logsign}^{[n]} X = l$).

The solution of this problem is in general not unique. For example, one can look for a solution that is piecewise linear. It is easy to see that the problem of finding such a solution is equivalent to the problem of solving a system of polynomial equations. Provided that there exists a solution this can be done for example by using Gröbner bases. In some sense, the remaining problem is to relate the level of truncation and the number of linear pieces in such a way that a solution exists. A disadvantage of this approach is a high complexity of the problem of solving polynomial systems.

In this paper we describe an algorithm that solves the reconstruction problem approximately. Namely, for given truncated signature $g \in G^{[n]}$, we construct a path $X$ such that $\mathrm{sign}^{[n]} X$ is approximately the same as $g$ (in a sense to be made precise). The algorithm is "fractal" and the idea is analogous that of the decimal (or binary) representation of the real numbers where one finds a good approximation (by $0.0, 0.1, \cdots, 0.9$) subtracts it, multiplies the remainder by ten and approximates again. Roughly speaking, we first fix a finite subset of $G^{[n]}$ (which is called the database and plays the same role as the numbers $0.0, 0.1, \ldots, 0.9$ for the decimal representation) and construct paths with corresponding signatures using any (maybe very slow and complicated) reconstruction method. This database must be computed only once and then it will be used for constructing paths with arbitrary truncated signatures. Such paths will be constructed using the scaling and the concatenation of the paths corresponding to the truncated signatures from the database.

**4. Properties of the signatures.** Denote by $G$ (resp., $L$) the set of all elements which are the signatures (resp., logarithmic signatures) of paths of bounded variation. The remarkable fact about the group structure of $G$ was observed by Chen (see [1]): for two paths $X^1$, $X^2$ and their concatenation one has $\mathrm{sign}(X^1 \circ X^2) = \mathrm{sign} X^1 \otimes \mathrm{sign} X^2$. and if $Y$ is the path which is obtained by going backwards along a path $X$ then $\mathrm{sign} X \otimes \mathrm{sign} Y = 1$ and $\mathrm{logsign} Y = -\mathrm{logsign} X$. Further, there is a natural scaling $\delta_\lambda$ on $T((\mathbb{R}^d))$ given by

$$\delta_\lambda(g_0 + g_1 + g_2 + \ldots) = g_0 + \lambda g_1 + \lambda^2 g_2 + \ldots$$

It is easy to see that $\delta$ is compatible with the scaling of the path, i.e. $\mathrm{sign}(\lambda X) = \delta_\lambda(\mathrm{sign} X)$ and $\mathrm{logsign}(\lambda X) = \delta_\lambda(\mathrm{logsign} X)$.

Let $||\cdot||$ be a norm on $\mathbb{R}^d$. Denote by the same symbol norms on the linear spaces $(\mathbb{R}^d)^{\otimes k}$, $1 \le k \le n$, such that $||u \otimes v|| \le ||u||\,||v||$, and that the norms are equivalent under permutations of the indices of the tensors. We define a pseudo-norm on $T^{[n]}(\mathbb{R}^d)$ by

$$|||g||| = \max_{1 \le k \le n} \left(||g_k||\right)^{1/k}.$$

Obviously, it is compatible with the path scaling.

**5. Construction of the database.** Denote by $L^{(k)}$ the $k$-th grade of $L$ and $p = \dim L^{(k)}$. Consider the ball $B \subset \mathbb{R}^p$ of radius $R$ centered at zero and a tetrahedron $T_p \subset B$

[4] although at these low sampling rates it cannot achieve the same compression as MPEG.

with $p+1$ vertices centered at zero as well. Denote by $h$ the distance between a vertex of $T_p$ and zero. Further, consider the balls $B_i, 0 \le i \le p$ of radius $r$ centered at the vertices of $T_p$. Our goal is to cover $B$ by the balls $B_i$ of a possibly small radius. Denote by $r_p(h,R)$ the smallest suitable radius for fixed $h$ and $R$ and by $r_p(R) = \min_{h>0} r_p(h,R)$ the optimal radius.

**Lemma 1.** *Let $p > 1$. Then $B$ is covered by the sets $B_i$ if and only if*

$$r \ge r_p(h,R) = \left(h^2 - 2Rh/p + R^2\right)^{1/2} \tag{2}$$

*Further, $r_p(R) = R\sqrt{1-p^{-2}}$, which is attained at $h_p(R) = R/p$.*

**Proof.** Denote by $l$ the edge length of $T_p$, by $a$ the distance between zero and a facet, by $b$ the distance between the center of a facet and a vertex belonging to this facet, and by $c$ the distance between a facet and the intersection point of the spheres of the radius $r$ centered at the vertices belonging to this facet. Then we have

$$\begin{aligned} l &= h\sqrt{2(1+p^{-1})}, & b &= l\sqrt{(1-p^{-1})/2} = h\sqrt{1-p^{-2}}, \\ a &= h/p, & c &= \sqrt{r^2 - b^2} = \sqrt{r^2 - (1-p^{-2})h^2}. \end{aligned}$$

Further, the condition for the balls $B_i$ to cover $B$ can be written as $R \le a + c$, i.e.,

$$R \le h/p + \sqrt{r^2 - (1-p^{-2})h^2},$$

which is equivalent to (2) as we also have $l \le R$.

Further, since $r_p(h,R)$ is of the parabolic type with respect to $h$ it attains its minimum at $h_p(R) = R/p$ and the minimal value $r_p(R)$ is then just equal to $r_p(h_p(R),R)$. □

Let us take $R = 1$ and consider a corresponding optimal tetrahedron $T_p$. Since we can identify $\mathbb{R}^p$ with $L^{(k)}$ we can regard the vertices $h_k^i$, $1 \le i \le \dim L^{(k)}$, of $T_p$ as elements of $L^{(k)}$. Now we define the set $H^k$ as

$$H^k = \{h_k^i : 1 \le i \le \dim L^{(k)}\} \subset L^{(k)} \subset L^{[n]}.$$

The set $H = H^1 \cup \dots \cup H^n$ is called the database.

**6. Reconstructing a signal using the database.** Let $l \in L^{[n]}$ and the precision $\varepsilon$ be given. We construct a path with the logarithmic signature $l_\varepsilon$ such that $|||\log(e^{-l_\varepsilon} e^l)||| < \varepsilon$. The algorithm consists of $n$ steps. Set $l^{(0)} = l$. For $1 \le i \le n$

**Step *i*:** Define

$$l^{(i)} = \log(e^{-\delta(m_i^{k_i})h_i^{k_i}} \dots e^{-\delta(m_i^1)h_i^1} e^{l^{(i-1)}}),$$

where $h_i^j \in H_i$ and $m_i^j > 0$, $1 \le j \le k_i$, are such that $||l_i^{(i)}|| < \varepsilon^i$. We are doing this in the following way. First, we consider $||l_i^{(i-1)}||$. If it is less then $\varepsilon^i$ we take $l^{(i)} = l^{(i-1)}$ and go to the next step of the algorithm. Otherwise we define $m_i^1 = ||l_i^{(i-1)}||^{1/i}$ and consider the element $\delta(1/m_i^1) l_i^{(i-1)}$ of the unit ball in $L^{(i)}$. By Lemma 1 and by the construction of $H_i$ there exists $h_i^1 \in H_i$ such that

$$\left|\left|\delta(1/m_i^1) l_i^{(i-1)} - h_i^1\right|\right| \le \sqrt{1 - d_i^{-2}}.$$

Assume that we have constructed $h_i^1, \dots, h_i^j$ and $m_i^1, \dots, m_i^j$ for some $j \ge 1$. Then we consider $||l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \dots - \delta(m_i^j)h_i^j||$. If it is less then $\varepsilon^i$ we take

$$l^{(i)} = \log \prod_{s=j}^{1} e^{-\delta(m_i^s)h_i^s}$$

and go to the next step of the algorithm. Otherwise we define

$$m_i^{j+1} = ||l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j||^{1/i}$$

and consider the element $\delta(1/m_i^{j+1})(l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j)$ of the unit ball in $L^{(i)}$. Again by Lemma 1 and by the construction of $H_i$ there exists $h_i^{j+1} \in H_i$ such that

$$\left|\left|\delta(1/m_i^{j+1})(l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j) - h_i^{j+1}\right|\right| \leq \sqrt{1 - d_i^{-2}}.$$

Notice that we will get $l^{(i)}$ in finitely many steps $k_i$ since we have

$$||l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j|| \leq ||l_i^{(i-1)}|| \left(1 - d_i^{-2}\right)^{j/2} \to 0 \text{ as } j \to \infty.$$

We have $l_j^{(i)} = l_j^{(i-1)}$ for all $j < i$, which implies $||l_j^{(i)}|| < \varepsilon^j$ for all $j \leq i$. In particular, $\operatorname{dist}(l^{(n)}) < \varepsilon$, the corresponding signature $l_\varepsilon$ is defined by

$$l_\varepsilon = \log \prod_{i=n}^{1} \prod_{j=k_i}^{1} e^{-\delta(m_i^j)h_i^j},$$

and a path with such logarithmic signature can be now constructed as the concatenation of the scaled paths from the database.

**7. Quality of the algorithm.** There are two quantities characterizing the quality of the algorithm: the size $N_d(n)$ of the database and the number of computations $K_d(n)$ required for reconstructing a path.

The exact expression for $N_d(n)$ is given by the Witt formula (see [3])

$$N_d(n) = \sum_{k=1}^{n} (d_k + 1) = \sum_{k=1}^{n} \sum_{i|k} \frac{\mu(i)d^{k/i}}{k} + n,$$

where $\mu$ is the Möbius function. For large $n$ and $d$ we have $N_d(n) \sim d^n/n$.

**Lemma 2.** *Let $l, h \in L^{[n]}$ such that $|||h||| \leq |||l|||$. Then $||| \log(e^{-h}e^l)||| \leq (ne^3)|||l|||$.*

**Proof.** Denote by $[\cdot]_s : L^{[n]} \to L^{(s)}$ the natural projection. Using the Dynkin formula (see [3]) and the inequalities $||l_k|| \leq |||l|||^k$ and $|||h_k|||^k \leq |||l|||$ following from the assumptions of the lemma, we obtain after some computations

$$||[\log(e^{-h}e^l)]_s|| \leq |||l|||^s s^s \sum_{k=1}^{s} \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{m_1+\cdots+m_{2k}=m} \frac{m!}{m_1! \ldots m_{2k}!} \leq |||l|||^s s^s \sum_{k=1}^{s} e^{2k} \leq (|||l|||se^3)^s,$$

which implies the desired estimate $||| \log(e^{-h}e^l)||| \leq |||l||| \max_{i \leq s \leq n} se^3 = (ne^3)|||l|||$. □

For every $l \in L^{[n]}$ and $\varepsilon > 0$, denote $G_\varepsilon(l) = \log(|||l|||/\varepsilon)$.

**Lemma 3.** *Let $l \in L^{[n]}$ and $\varepsilon$ be given. Then $k_i \leq G_\varepsilon(l) 2d^{i(i+1)}(3\log(ne^3))^{i-1}/i!$ for all $i$.*

**Proof.** The number $k_i$ of iterations we are required to do at the step $i$ is the maximal natural number such that the inequality $||l_i^{(i-1)}|| \left(1 - d_i^{-2}\right)^{k(i)/2} > \varepsilon^i$ holds. This means that

$$k_i = \left[\frac{2\log(\varepsilon^i/||l_i^{(i-1)}||)}{\log(1 - d_i^{-2})}\right] \leq 2id_i^2 \log \frac{\operatorname{dist}(l^{(i-1)})}{\varepsilon} = 2id_i^2 G_\varepsilon(l^{(i-1)}), \tag{3}$$

where we used the inequality $-\log(1-x) \geq x$. Let us now estimate $G_\varepsilon(l^{(i)})$, which itself depends on $k_i$. By Lemma 2, using (3) and the inequality $d_i < d^i/i$, we obtain

$$G_\varepsilon(l^{(i)}) \leq \log \frac{(ne^3)^{k_i} \operatorname{dist}(l^{(i-1)})}{\varepsilon} = k_i \log(ne^3) + G_\varepsilon(l^{(i-1)}) \leq G_\varepsilon(l^{(i-1)})(2id_i^2 \log(ne^3) + 1)$$
$$\leq G_\varepsilon(l) \prod_{j=1}^{i} (2jd_j^2 \log(ne^3) + 1) \leq G_\varepsilon(l) \prod_{j=1}^{i} \frac{3d^{2j}}{j} \log(ne^3) \leq G_\varepsilon(l) \frac{(3\log(ne^3))^i d^{i(i+1)}}{i!}$$

Combining this inequality with (3) we get the required estimate. □

Denote by $E_d(n)$ (resp., by $P_d(n)$) the number of computations, which is sufficient for computing exp or log (resp., the product of two elements) in $\mathbf{L^{[n]}}$. Further, denote by $K_d^i(n)$ the number of computations required to perform the $i$-th step of the algorithm.

**Lemma 4.** *For every $1 \leq i \leq n$ one has $K_d^i(n) \leq 2k_i(d_i+1)^2 + (k_i+2)E_d(n) + k_i P_d(n)$.*

**Proof.** Consider the $j$-th iteration of the $i$-th step. We need $2d_i+4$ computations to compute $||m_i^j||$ and to compare it with $\varepsilon$. Further, we need $d_i+1$ computations for $\delta(1/m_i^{j+1})(l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j)$ and $2(d_i+1)^2$ computations to find $||\delta(1/m_i^{j+1})(l_i^{(i-1)} - \delta(m_i^1)h_i^1 - \cdots - \delta(m_i^j)h_i^j) - h||$ for all elements of $H_i$. Finally, we need $d_i+1$ computations for choosing an appropriate $h \in H_i$. So the total number of computations for the $j$-th iteration is $2(d_i+2)^2$, and there are $k_i$ iterations. It remains to notice that in conclusion of the $i$-th step we compute $l^{(i)}$, which requires at most $(k_i+2)E_d(n) + k_i P_d(n)$ computations. □

By Lemma 3 we have

$$\sum_{i=1}^{n} k_i \leq \sum_{i=1}^{n} G_\varepsilon(l) \frac{2d^{i(i+1)}(3\log(ne^3))^{i-1}}{i!} \leq 2G_\varepsilon(l) d^{n^2+n} (3\log(ne^3))^{n-1}$$
$$\sum_{i=1}^{n} k_i(d_i+1)^2 \leq \sum_{i=1}^{n} G_\varepsilon(l) \frac{2d^{i(i+1)}(3\log(ne^3))^{i-1}}{i!} \left(\frac{d^i}{i}+1\right)^2 \leq 5G_\varepsilon(l) d^{n^2+3n} (3\log(ne^3))^{n-1}.$$

Denote by $\varphi(n)$ the rate of growth of $E_d(n)$ and $P_d(n)$, i.e., $E_d(n), P_d(n) \ll d^{\varphi(n)}$. It follows now from Lemma 4 and the previous estimates that

$$K_d(n) = \sum_{i=1}^{n} K_d^i(n) \leq 2G_\varepsilon(l) d^{n^2+n} (3\log(ne^3))^{n-1}(5d^{2n} + E_d(n) + P_d(n)) + 2nE_d(n) \ll d^{n^2+f(n)}$$

which gives the asymptotic upper bound for the speed of the algorithm.

# References


[1] Chen, K.-T., Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula, Ann. of Math. 65, 163-178, 1957

[2] Lyons, T.J., Differential equations driven by rough signals. Rev. Mat. Iberoamericana 14, no. 2, 215-310, 1998

[3] Reutenauer, C., Free Lie Algebras. London Mathematical Society Monographs, New Series 7, Oxford Science Publications, 1993

[4] Hambly, B.H. and Lyons, T.J.: Uniqueness for the signature of a path of bounded variation and continuous analogues for the free group. Preprint.